UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/710,218	11/10/2000	Timothy L. Harris	1004-4896	4731

22120 7590 06/02/2004

ZAGORIN O'BRIEN & GRAHAM, L.L.P.
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2126	7

DATE MAILED: 06/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/710,218	HARRIS, TIMOTHY L.
	Examiner	**Art Unit**
	Li B. Zhen	2126

PRL

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 29 are pending in the application.

Response to Arguments

2. Applicant's arguments filed March 16, 2004 have been fully considered but they are not persuasive.

3. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "if a data structure is linearizable, then 'the data structure behaves as if the operations on it requested by various processes are performed atomically in some sequential order and that such operations appear to take effect at some point between their invocation and response," p. 3, lines 15 – 18) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Therefore, a reasonable interpretation of the limitation "linearizable operations" will be atomic operations because an atomic operation either runs from beginning to end without interruption or it is aborted.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

5. **Claims 1, 16, 25 and 29 are rejected under 35 U.S.C. 102(a) as being anticipated by "DCAS-Based Concurrent Deques" (hereinafter Agesen).**

6. As to claim 16, Agesen teaches managing access to a linked-list of nodes [a linked-list representation, and is the first non-blocking unbounded-memory deque implementation; p. 137, Abstract] susceptible to concurrent operations on a group [allows uninterrupted concurrent access to both ends of the deque; p. 137, Abstract], comprising:

separating deletion of a value [pop operation] from the group into at least two functional sequences [splitting the pop operation into two steps, p. 138, Section 1.2; a node is always removed...in two separate, atomic steps, p. 141, Section 4];

the first function sequence performing a logical deletion of the value using a synchronization primitive [first the node is "logically" deleted, by replacing its value with null and setting a special deleted bit to 1 in the sentinel to indicate the presence of a logically deleted node; p. 141, Section 4] to mark a corresponding one of the nodes [marking that a node is about to be deleted; p. 138, Section 1.2]; and

the second functional sequence excising the marked node [the node is "physically deleted, by modifying pointers so that the node is no longer in the doubly-linked chain of nodes; p. 141, Section 4] from the linked-list [once marked, the node is considered "deleted," and the actual deletion from the list can then be performed by the next push or next pop operation on that side of the deque; p. 138, Section 1.2].

7. As to claim 1, Agesen teaches a non-blocking concurrent shared object representation [a linked-list representation, and is the first non-blocking unbounded-memory deque implementation; p. 137, Abstract]:

a linked-list of nodes encoding of a group of zero or more values [represent a deque as a doubly-linked list. Each node in the list contains two link pointers and a value; p. 141, Section 4]; and

linearizable operations [require that a concurrent implementation of a deque object be one that is linearizable, p. 139, Section 2.2] defined to implement insert [pushRight(), pushLeft(); p. 139, Section 2.2] and remove operations [popRight(), popLeft(); p. 139 Section 2.2] on the group, wherein concurrent execution of the linearizable operations [by being linearizable and non-blocking, our concurrent deque implementations are guaranteed to behave as if operations on them were executed in a mutually exclusive manner; p. 137, Section 1] is mediated using a first synchronization primitive [remove the left node by performing a DCAS on the left sentinel's pointer and the right node's L pointer; p. 144, left col.] to encode a marked node indication signifying logical deletion of a corresponding one of the values from the group [first the node is "logically" deleted, by replacing its value with null and setting a special deleted bit to 1 in the sentinel to indicate the presence of a logically deleted node; p. 141, Section 4].

8. As to claim 25, this is a product claim that corresponds to claim 1; note the rejection to claim above, which also meets this product claim.

9. As to claim 29, this is an apparatus claim that corresponds to method claim 16; see the rejection to claim 16 above, which also meets this apparatus claim. As to the additional limitations, Agesen teaches plural processors [concurrent system consists of a collection of n processors; p. 138, Section 2], one or more data stores [shared data structure called objects; p. 138, Section 2], and means for traversing the encoded group without use of an atomic operation [Read(L) reads location L and returns its value; p. 138, Section 2].

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. **Claims 1 – 8, 10 – 22 and 25 – 29 are rejected under 35 U.S.C. 102(b) as being anticipated by "A Lock-Free Multiprocessor OS Kernel" (hereinafter Massalin).**

12. As to claim 16, Massalin teaches managing access to a linked-list of nodes susceptible to concurrent operations on a group [Compare-and-Swap to perform linked-list insert and delete from the head of the list, Section 3.3, p. 5, second paragraph; in the middle of the list, we can achieve the same effect by deleting a node only when the permanence of the previous node is guaranteed; Section 3.3, p. 6], comprising:

separating deletion of a value from the group into at least two functional sequences [do this in two steps: (1) mark the nodes to be deleted and leave them in the list (2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3];

the first function sequence performing a logical deletion of the value using a synchronization primitive [set the mark at the mark at the same time we enter the node using a two-word Compare-and-Swap; Section 3.3, p. 6] to mark a corresponding one of the nodes [(1) mark the nodes to be deleted and leave them in the list; p. 6, Section 3.3]; and

the second functional sequence excising the marked node from the linked-list [(2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

13. As to claim 1, Massalin teaches non-blocking [Compare-and-swap is the foundation of lock-free synchronization; Section 2.3, p. 2] concurrent shared object representation [Two-word Compare-and-Swap lets us efficiently implement many basic data structures such as stacks, queues, and linked lists because we can atomically update both a pointer and the location being pointed to; Section 2.3, p. 2]:

a linked-list of nodes encoding of a group of zero or more values [Section 3.3, General Linked Lists, p. 5]; and

linearizable operations defined to implement insert and remove operations on the group [Compare-and-Swap to perform linked-list insert and delete from the head of the

list, Section 3.3, p. 5, second paragraph; in the middle of the list, we can achieve the same effect by deleting a node only when the permanence of the previous node is guaranteed; Section 3.3, p. 6], wherein concurrent execution of the linearizable operations is mediated using a first synchronization primitive [set the mark at the mark at the same time we enter the node using a two-word Compare-and-Swap; Section 3.3, p. 6] to encode a marked node indication signifying logical deletion of a corresponding one of the values from the group [(1) mark the nodes to be deleted and leave them in the list (2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

14. As to claim 25, this is a product claim that corresponds to claim 1; note the rejection to claim above, which also meets this product claim.

15. As to claim 29, this is an apparatus claim that corresponds to method claim 16; see the rejection to claim 16 above, which also meets this apparatus claim. As to the additional limitations, Massalin teaches plural processors [multiprocessor OS kernel using only lock-free synchronization methods based on Compare-and-Swap; see abstract], one or more data stores [quajects, chunks of code with data structures; Section 3, p. 3], and means for traversing the encoded group without use of an atomic operation [separate the run-queue traversal...from the queue element update; p. 3, 4th full paragraph].

16. As to claim 2, Massalin teaches physically excise the node corresponding to the logically deleted value [if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

17. As to claim 3, Massalin teaches compare and swap (CAS) operations [set the mark at the mark at the same time we enter the node using a two-word Compare-and-Swap; Section 3.3, p. 6].

18. As to claim 4, Massalin teaches reclamation of storage associated with the excised node is independent of the linearizable operations [separate the run-queue traversal (done lock-free, safely and concurrently) from the queue element update (done locally); p. 3, 4th full paragraph].

19. As to claim 5, Massalin teaches the linked-list of nodes is free of reference count storage for coordination of garbage collection [simplify the implementation to use a binary marker instead of counters; p. 6, 1st full paragraph].

20. As to claim 6, Massalin teaches traversal of the concurrent shared object is without atomic update of a garbage collection coordination store [separate the run-queue traversal (done lock-free, safely and concurrently) from the queue element update (done locally); p. 3, 4th full paragraph].

21. As to claim 7, Massalin teaches successful completion of an insertion into the group requires at most one atomic update of the concurrent shared object [insert reads the address of the list's first element into a private variable...copies it into the link field of the new element to be inserted, and then uses Compare-and-Swap to atomically update the list's head pointer if it had not been changed since the initial read; Section 3.3, p. 5], successful completion of a deletion from the group requires, at most, two atomic updates of the concurrent shared object [do this in two steps: (1) mark the nodes to be deleted and leave them in the list (2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3], and traversal of the concurrent shared object is without atomic update of the concurrent shared object [separate the run-queue traversal (done lock-free, safely and concurrently) from the queue element update (done locally); p. 3, 4th full paragraph].

22. As to claim 8, the node corresponding to the logically deleted value is physically excised from the linked-list by an execution sequence corresponds to one of:

an instance of the remove operation that performed the logical deletion [since step 2 may require going back the list an arbitrary number of nodes, usually we do the step 2 the next time we traverse the list to avoid the overhead of traversing the list just for deletion; p. 6, Section 3.3];

an instance of the remove operation that did not performed the logical deletion [if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3]; and

an instance of the insert operation [insert reads the address of the list's first element into a private variable... uses Compare-and-Swap to atomically update the list's head pointer if it had not been changed since the initial read; Section 3.3, p. 5].

23. As to claim 10, Massalin teaches the values of the group are stored in respective ones of the nodes [if the shared data is larger than two words, then we try to encapsulate it in on of the lock-free objects; p. 3, 2nd full paragraph].

24. As to claims 11 and 12, Massalin teaches the values of the group are represented in storage identified by respective ones of the nodes [read the current value of the stack pointer into a private variable, de-reference it to get the top item on the stack, and increment the stack pointer into a private variable; Section 3.1, p. 4].

25. As to claim 13, Massalin teaches the marked node indication includes a distinguishing pointer value [simplify the implementation to use a binary marker instead of counters; p. 6, 1st full paragraph].

26. As to claim 14, Massalin teaches the marked node indication includes a distinguishing bit value [simplify the implementation to use a binary marker instead of counters; p. 6, 1st full paragraph] in an otherwise unused portion of a next node pointer of the logically deleted node [set the mark at the time we enter the node using a two-word Compare-and-Swap; p. 6, 1st full paragraph].

27. As to claim 15, Massalin teaches respective next node pointers of those of the nodes corresponding to current values of the group directly reference respective other ones of the nodes [delete operation is "safe" if the deleted node's link pointers continue to be valid, i.e., pointing to nodes that eventually take it back to the main list where the Compare-and-Swap will detect the change; Section 3.3, p. 6], and the marked node indication includes a distinguishing additional level of indirection between the next node pointer of the logically deleted node and a respective other one of the nodes [two-word Compare-and-Swap can guarantee safety by simultaneously checking the previous node's pointer; Section 3.3, p. 6].

28. As to claims 17, 19 and 20, this is rejected for the same reasons as claim 8 above.

29. As to claim 18, Massalin teaches the logical deletion is performed as part of a deletion operation operating upon the value [(1) mark the nodes to be deleted and leave them in the list; p. 6, Section 3.3], and the marked node excision is performed as part of another operation [(2) if the previous node is not marked for deletion, sit on it and delete the original node marked for deletion; p. 6, Section 3.3].

30. As to claim 21, this is rejected for the same reasons as claim 3 above.

31. As to claim 22, Massalin teaches after the logical deletion but before the marked node excision, traversing, as part of an access operation, the linked-list including the marked node [since step 2 may require going back the list an arbitrary number of nodes, usually we do the step 2 the next time we traverse the list to avoid the overhead of traversing the list just for deletion; p. 6, Section 3.3].

32. As to claims 26 and 27, see the rejection to claim 1 above.

33. As to claim 28, Massalin teaches at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium [2.5 MB no-wait state main memory, 390 MB hard disk; p. 17, Section A.2].

Claim Rejections - 35 USC § 103

34. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

35. **Claims 9, 23 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Massalin in view of U.S. Patent NO. 6,581,063 to Kirkman.**

36. As to claim 9, Massalin teaches managing access to a linked-list of nodes [Section 3.3, p. 5 – 6], but does not teach a find operation.

However, Kirkman teaches linked-list maintaining including an insert operation [col. 6, lines 40 – 63], a delete operation [remove function; col. 6, line 64 – col. 7, line 20] and a find operation [searching is necessary, there are not an excessive number of inactive objects to search through; col. 72, lines 15 – 35].

37. It would have been obvious to a person of ordinarily skilled in the art at the time of the invention to apply the teaching of a find operation as taught by Kirkman to the invention of Massalin this provides an pre-defined search capability that returns a requested element of the linked list without additional code for traversing the link-list to locate the requested element.

38. As to claims 23 and 24, Massalin as modified the ordered set is organized in increasing value order [link may contain a textual data field such as a name, where the links are to be ordered such that the textual data fields are arranged in alphabetical order; col. 76, lines 13 – 27 of Kirkman], and the remove operation is selective for a value, if any, of the group greater than or equal to the specified value [find the link which is yet to be removed...The link which is to be removed is located; col. 27, lines 40 – 67 of Kirkman].

Conclusion

39. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent NO. 5,924,098 to Kluge teaches a method of managing a linked-list data structure maintained and accessed by a computer program.

U.S. Patent NO. 6,360,220 to Forin teaches a lock-free method for accessing and storing information in an indexed computer data structure having modifiable entries.

40. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (703) 305-3406. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2126

lbz
May 18, 2004

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100